



REC'D 18 AUG 2003

WIPO

PCT

**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

Aktenzeichen:

102 25 099.5

Anmeldetag:

05. Juni 2002

Anmelder/Inhaber:

Systemonic AG, Dresden/DE

Bezeichnung:

Verfahren zum codieren/decodieren von VLIW
cached Befehlen

IPC:

G 06 F, 9/445

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 30. Juni 2003
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Jerofsky

**LIPPERT, STACHOW, SCHMIDT & PARTNER**

Patentanwälte · European Patent Attorneys · European Trademark Attorneys

Krenkelstraße 3 · D-01309 Dresden

Telefon +49 (0) 3 51.3 18 18-0

Telefax +49 (0) 3 51.3 18 18 33

Ad-wb/wb

5. Juni 2002

5 **Systemonic AG**
01099 Dresden

10 **Verfahren zum codieren/decodieren von VLIW cached Befehlen**

Zusammenfassung

15 Die Erfindung betrifft ein Verfahren zur Ansteuerung von Funk-
tionseinheiten in einem Prozessor, wobei in einer Konfigura-
tionsphase eine aus einer Übersetzung eines Programmcodes stam-
mende Folge von Primärinstruktionsworten eine Fraktionierung in
eine Folge von Instruktionswortteile erfährt, wobei in einem
20 Programmablauf den Prozessor ansteuernde Instruktionsworte in
der vollen Instruktionswortbreite erzeugt und in einem Instruk-
tionswortspeicher (Cache) zwischengespeichert werden.

Die erfindungsgemäße Aufgabenstellung, in der Ausführungsphase
eine Steigerung der Prozessorperformance durch Erhöhung des
25 Komprimierungsgrad der Primärinstruktionsworte in ihre fraktio-
nierten Instruktionswortteilen unabhängig von speziellen Merk-
malen (z. B. Periodizität) der FIW (Funktions-Instruktions-
Wortteile) zu erreichen, wird dadurch gelöst, dass in der Kon-
figurationsphase in einem ersten Schritt die Fraktionierung ei-
30 nes Primärinstruktionswortes in ein TVLIW (Tagged Very Long In-
struction Word) erfolgt und in einem zweiten Schritt wird je-
weils das vorliegenden TVLIW in ein HVLIW (Headed Very Long In-
struction Word) gewandelt. In dem HVLIW wird ein General Header
vorangestellt. Das HVLIW ersetzt mit seiner codeverdichteten
35 Struktur alle Funktionen des TVLIW (1). (Fig.1)

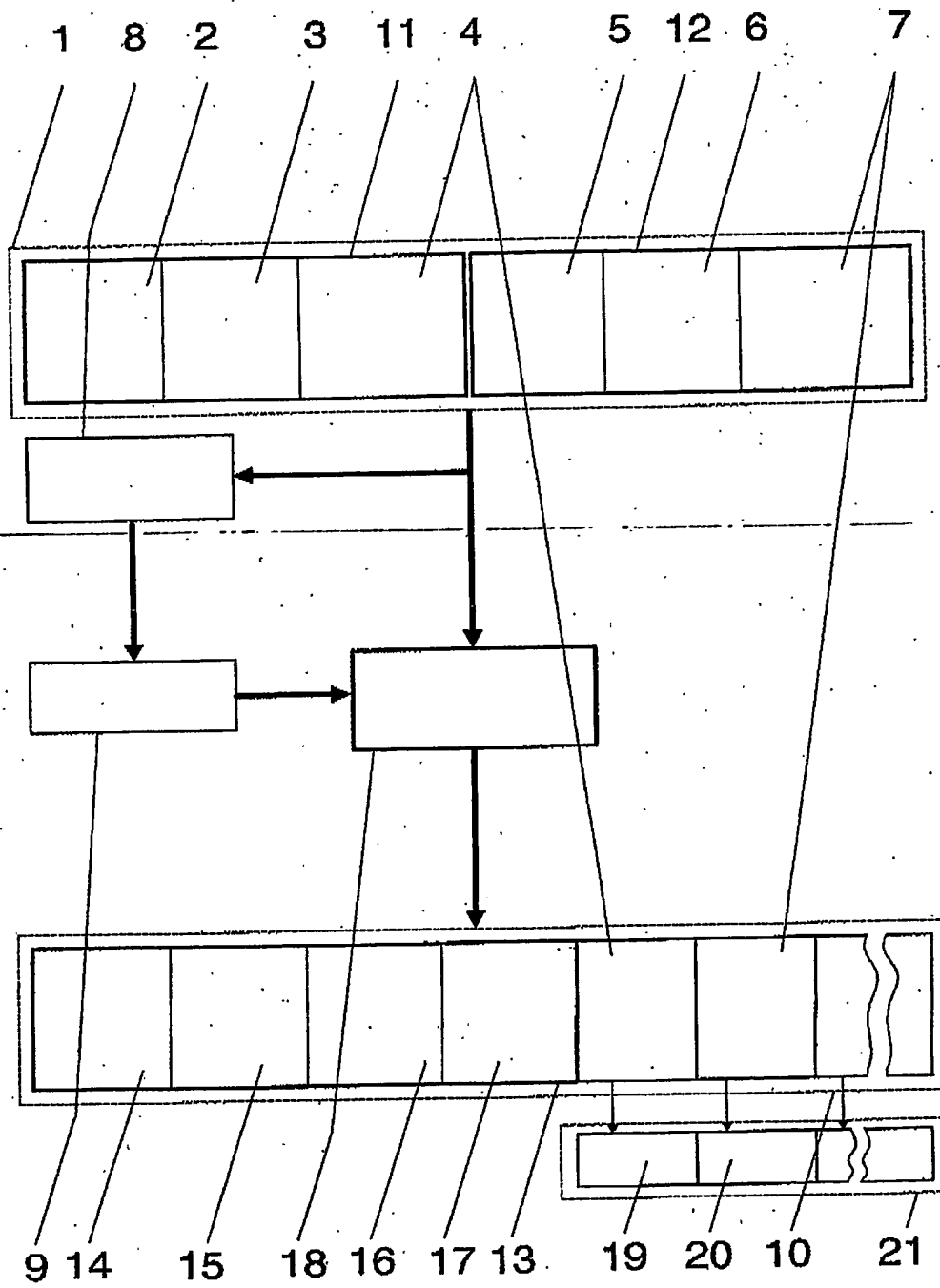


Fig.1

LIPPERT, STACHOW, SCHMIDT & PARTNER

Patentanwälte · European Patent Attorneys · European Trademark Attorneys

Krenkelstraße 3 · D-01309 Dresden

Telefon +49 (0) 3 51.3 18 18-0

Telefax +49 (0) 3 51.3 18 18 33

Ad-wb/wb

5. Juni 2002

5 **Systemonic AG****01099 Dresden**10 **Verfahren zum codieren/decodieren von VLIW cached Befehlen**

Die Erfindung betrifft ein Verfahren zur Ansteuerung von Funktionseinheiten in einem Prozessor, wobei in einer Konfigurationsphase eine aus einer Übersetzung eines Programmcodes stammende Folge von Primärinstruktionsworten eine Fraktionierung in eine Folge von Instruktionsteilen erfährt und in einem Programmablauf nachfolgend den Prozessor ansteuernde Instruktionsteile in der vollen Instruktionsteilbreite als VLIW (Very Long Instruction Word) erzeugt und in einem Instruktionsteilspeicher (Cache) zwischengespeichert werden.

Hierzu sind verschiedene Lösungen, die eine jeweils vorteilhafte Variante zur Synthese eines VLIW (Very Long Instruction Word) aus den im Programmablauf auftretenden Instruktionsteilen behandeln, bekannt.

Ihnen ist gemeinsam, dass die Primärinstruktionsteile, welche aus einer Übersetzung des Programmcodes resultieren, als Folge fraktionierter Instruktionsteile generiert werden.

30 So erfolgt der Aufbau eines aktuellen VLIW aus einer begrenzten Anzahl von Funktionsinstruktionsteilen (FIW), wobei jedes dieser FIW genau eine Funktionseinheit (FU) des Prozessors ansteuert.

35 Die Deutsche Patentschrift DE 198 59 389 C1 charakterisiert den Stand der Technik für Verfahren der eingangs genannten Art.

Bei dieser Lösung erfolgt die Fraktionierung der im Programm vorliegender Primärinstruktionsworte in einzelnen Programmworten, welche auch vorteilhaft als TVLIW (Tagged Very Long Instruction Word)-Container bezeichnet werden.

TVLIW-Container deshalb, weil dem einzelnen Programmwort neben einem Informationsteil, der vorwiegend durch ein FIW (Funktion sinstruktionswort) repräsentiert wird, auch die Angaben über die Schreib- und Lesezeilennummer eines einzusetzenden Instruktionswortspeichers, zugehörig ist. Letztere Angaben stellen eine Kennung (Tag) zum FIW dar.

Außerdem gehören zum Programmwort auch die Angaben, wie mit dem jeweiligen Inhalt des so gekennzeichneten Instruktionswortspeichers zu verfahren ist, und werden somit durch eine Operationscode (Opcode) repräsentiert.

Bei dem o.g. Verfahren werden in günstiger Weise die Datenbreite des im Prozessor zu verarbeitenden Programmmes verringert, um den Hardwareaufwand und damit die Kosten für die Realisierung des jeweiligen Prozessors niedrig zu halten.

Weiterhin sind verschiedene Lösungen bekannt, die eine jeweils vorteilhafte Variante zur Synthese eines VLIW (Very Long Instruction Word) aus den im Programmablauf auftretenden FIW behandeln.

So ist durch die Druckschrift 102 03 541.5 zur deutschen Patent-Anmeldung der weiterführende Stand der Technik umrissen.

Bei diesem wird die in einer Konfigurationsphase durchgeführte Fraktionierung der Primärinstruktionsworte durch eine anschließende verfahrensmäßige maschinelle Ähnlichkeits-Analyse erweitert, in deren Folge, die mit bestimmten Ähnlichkeitsmerkmalen (periodische Eigenschaft) ausgewählten und somit mehrfach verwendbaren Instruktionswortteile zusammengefasst werden.

Diese Folge von Instruktionswortteilen wird mit einem diesbezüglich gemeinsamen Operationscode und einer für alle Glieder der Folge gültigen Kennzeichnung ihrer periodischen Eigenschaft, die mit der Anzahl der Glieder ergänzt wird, in der anschließenden Verarbeitungsphase zur Erzeugung des VLIW verwendet.

Auf diese Weise wird bei diesem speziellen Komprimierungsvorgang in der Konfigurationsphase die Auswahl und Kennzeichnung der Instruktionswortteile vorgenommen, die in der Ausführungsphase zur Zwischenspeicherung vorgesehen sind und damit bei der Wiederverwendung gleicher Instruktionswortteile Prozessor-Performance sparen.

Bei der zunehmenden Komplexität der Prozessoren und den Anforderungen an die Verarbeitungsgeschwindigkeit wird ersichtlich, dass eine höhere Kompression bei der Codierung der Instruktionswortteile und deren Decodierung zur Erzeugung des VLIW (Very Long Instruction Word) erreicht werden muss, da eine anderweitige Erhöhung der Verarbeitungsgeschwindigkeit, z.B. durch eine Erhöhung der Arbeitstaktfrequenzen, an physikalische Grenzen stößt.

Aufgabe der Erfindung ist es, in der Ausführungsphase eine Steigerung der Prozessorperformance durch Erhöhung des Komprimierungsgrads der Primärinstruktionswörter in ihre fraktionierten Instruktionswortteilen unabhängig von speziellen Merkmalen (Periodizität) der FIW zu erreichen.

Die erfindungsgemäße Aufgabenstellung wird dadurch gelöst, dass in einem ersten Schritt die Fraktionierung eines Primärinstruktionswortes in der Konfigurationsphase, in die Folge einer bestimmten Anzahl von Instruktionswortteilen, die in der Ausführungsphase für den Aufbau eines jeweiligen VLIW verwendet werden, erfolgt.

Dabei wird einem jeweiligen ersten und zweiten FIW (Funktions-Instruktionswortteil) ein ihm zugehöriger erster oder zweiter Operationscode vorangestellt. Dieser bestimmt damit, wie in der

Ausführungsphase mit dem durch das jeweilige FIW belegten Speicherplatz des Cache umgegangen wird.

Weiterhin anschließt sich dem jeweiligen ersten oder zweiten
5 Operationscode eine zugehörige erste oder zweite Kennung an, die die Information darstellt, welche erste oder zweite FU das jeweilige FIW ansteuert.

Es werden jeweils der erste bzw. der zweite Operationscode und
10 ihre zugehörige erste bzw. zweite Kennung mit den jeweiligen ersten bzw. zweiten FIW zu dem ersten bzw. zweiten TVLIW Container zusammengefasst.

Hierbei repräsentieren diese in ihrer Gesamtheit das TVLIW.

15 In einem zweiten Schritt wird in der Konfigurationsphase das jeweils vorliegenden TVLIW in ein HVLIW gewandelt. In dem HVLIW wird ein General Header vorangestellt.

20 Bei der Wandlung des TVLIW in das HVLIW gilt, dass dieses mit seiner enthaltenen codeverdichteten General Header Struktur alle Funktionen des TVLIW ersetzt.

In einer Variante wird die erfindungsgemäße Aufgabenstellung
25 dadurch gelöst, dass eine Betriebsart "Command Code" des HVLIW und seines zugehörigen General Headers realisiert wird. In diesem General Header ist die Information codiert abgelegt, die alle Kombinationen angibt, welches erste und zweite FIW (Instruktionswortteil), nach Decodierung, in der Ausführungsphase,
30 zur Ansteuerung einer jeweiligen ersten und/oder zweiten FU (Funktionseinheit) des Prozessors vorgesehen ist.

Weiterhin ist im General Header abgelegt, welches erste und/oder zweite FIW Speicherplätze des Cache belegt und ob bzw.
35 welche Operation mit dem jeweiligen Speicherinhalt beim Aufbau des VLIW in der Ausführungsphase im Cache ausgeführt werden sollen.

Diese Lösung zielt darauf ab, dass in der Betriebsart "Command

Code" des HVLIIW durch eine Zusammenfassung von mehreren FIW und einer dazu gehörigen Zusammenfassung der Angaben, welches der FU von welchem FIW anzusteuern ist, sowie welche FIW beim Aufbau des VLIW bestimmte Speicherplätze des Cache belegt und welche Operation dann mit deren Speicherinhalt bezüglich anderer Speicherplätze des Cache ausgeführt werden, die angestrebte Kompression der Instruktionen realisiert wird.

10 Auf diese Weise werden Speicherplatz gespart und Prozessorperformance geschont.

Es wird eine vorteilhafte Ausprägung der Variante der erfindungsgemäßen Lösung der Aufgabenstellung dadurch erreicht, dass in dem ersten Teil des General Headers ein Header Mode, welcher eine Information über die Betriebsart "Command Code" des HVLIIW und des General Headers enthält, bereitgestellt wird.

20 Diesem anschließt sich ein zweiter Teil an, in dem die jeweilige meist gebrauchte Kombination, welche der jeweiligen FU durch welches erstes oder zweites FIW angesteuert wird, enthalten ist.

25 Diese meist gebrauchte Kombination ist als codierter Tabellenwert im Wörterbuch abgelegt.

Es wird ein dritter Teil als CE-Information (Cache-Extra-Information) angeschlossen, in dem ein Zeiger, der auf eine vorgesehene Stelle des Wörterbuches verweist, enthalten ist.

30 Als letzter Teil des General Headers ist die Zusatzinformation vorgesehen.

35 In unmittelbarer Reihenfolge an den General Header schließen sich die für den Aufbau des VLIW notwendigen ersten und zweiten FIW an.

Diese erfindungsgemäße Lösung hebt darauf ab, für die Betriebsart "Command Mode" einen strukturierten General Header bereit-

zustellen, der sehr flexibel ist und alle Arten von "Command Code" unterstützt. Dieser soll auch für Weiterentwicklungen und update seine Gültigkeit behalten und seine Kompressionsmöglichkeiten absichern.

5

Eine weitere Variante der erfindungsgemäßen Lösung der Aufgabenstellung dadurch erreicht, dass eine Betriebsart "Referenz-Instruktion" des General Headers realisiert wird, bei der generell die zum Aufbau des VLIW in der Ausführungsphase vorgesehenen FIW im Cache zwischengespeichert werden.

10

Hierbei trägt der zugehörige Header Mode eine entsprechend decodierbare Kennung dieser Betriebsart "Referenz-Instruktion". Es wird die Betriebsart "Referenz-Instruktion" durch eine spe-

15

zielles HVLIW-eingeleitet.

In diesem ist eine Adressangabe enthalten, mit der auf eine Referenz-Instruktion verwiesen wird.

20

Weiterhin ist im nachfolgenden HVLIW, welches ebenfalls die Kennung der Betriebsart "Referenz-Instruktion" trägt, eine relative Adresse bezüglich der durch den Verweis vorliegenden Adresse enthalten.

25

Daran wird eine Maske angefügt, welche die von der Ansteuerung auszuschließenden FU repräsentiert.

30

Bei dieser günstigen Variante der erfindungsgemäßen Lösung werden durch die Realisierung der speziellen Betriebsart "Referenz-Instruktion" von HVLIW die großen Instruktionen für den Prozessor-Kernel vermieden, die z.B. auch im Header Mode "Command Code" groß ausfallen, weil sie für eine Vielzahl von FU (Funktionseinheiten) anwendbar sein müssen.

35

Dadurch sind auch jeweils Start- und Beendigungs-Phasen der Instruktionen zur Ansteuerung der Grundbestandteile der einzelnen FU notwendig.

Aufgrund vieler gleicher FIW, die zur Ansteuerung der FU in den Instruktionen z.B. in Digitalen Signal-Prozessoren (DSP) auftreten, ist aus der Kenntnis der Instruktionen für den Prozessor-Kernel ersichtlich, dass die jeweiligen Start- und Beendigungs-Phasen der Instruktionen für die jeweiligen FU redundant sind.

Diese Redundanz wird durch die erfindungsgemäße Lösung vermieden, in dem mittels des HVLIW, welches die Betriebsart "Referenz-Instruktion" einleitet, eine Adressangabe als einer Referenz vorgegeben wird.

Im nachfolgenden HVLIW der Betriebsart "Referenz-Instruktion" wird in seinem General Header nur eine relative Adresseangabe mitgeteilt, mittels der das erforderliche FIW in der Ausführungsphase decodiert werden kann.

Ebenfalls in diesem HVLIW wird, an den General Header anschließend, die erste und/oder zweite FU (Funktionseinheit), für die diese bestimmte Instruktion nicht verwendet werden soll, codiert angegeben.

Diese Angabe kann als Maske, welche die Ansteuerung von FU abschließen, wesentlich kürzer gestaltet werden als eine Angabe aller ansteuernden FIW.

Somit brauchen bei HVLIW in der Betriebsart "Referenz-Instruktion" die jeweiligen Start- und Beendigungs-Phasen des für die Ansteuerung der vorgesehenen FU bestimmten FIW im General-Header nur einmal angegeben werden.

Auf diese Weise wird Speicherplatz gespart.

Da mit dieser Kompression es nicht notwendig ist, die jeweiligen vollständigen Start- und Beendigungs-Phasen der Instruktionen beim VLIW Aufbau zu verarbeiten, wird folglich ebenfalls die Prozessorperformance in der Ausführungsphase auch weniger belastet.

5 Eine spezielle Variante der erfindungsgemäßen Lösung der Aufgabenstellung, welche die Betriebsart des General Headers "Referenz-Instruktion" lokal günstig realisiert, wird dadurch erreicht, dass das spezielle HVLIW, welches die Betriebsart "Referenz-Instruktion" einleitet, als enthaltene Adressangabe auf das vorhergehende HVLIW verweist.

10 Eine weitere spezielle Variante der erfindungsgemäßen Lösung der Aufgabenstellung, welche die Betriebsart des General Headers "Referenz-Instruktion" global günstig realisiert, wird dadurch erreicht, dass das spezielle HVLIW, welches die Betriebsart "Referenz-Instruktion" einleitet, als enthaltene Adressangabe auf eine allgemeine Adresse verweist.

15 Eine vorteilhafte Erweiterung zur Lösung der erfindungsgemäßen Aufgabenstellung speziell für die Betriebsart "Command Code" des HVLIW wird dadurch erreicht, dass in der Ausführungsphase das HVLIW in einem Decoder1 decodiert wird, welcher mit einem
20 Header-Decoder, einer CMTD (Command-Code-Decompression-Tabelle), einem Cache und einer Cache-Miss-Reparatur-Logik ausgestattet ist, wobei das HVLIW zwischengespeichert vorliegt.

25 Der Header-Decoder erkennt die Betriebsart "Command Code" des General-Headers aus dem dort gespeicherten Header Mode.

30 Weiterhin werden entsprechend des erkannten Header Mode die im General Header vorliegenden Werte der FU-C durch Vergleich mit der CMTD und in Verbindung mit der ebenfalls aus dem General Header entnommenen CE-Information dekomprimiert.

Entsprechend des erkannten Header Modes wird die Zusatzinformation des General Headers verarbeitet.

35 Ein möglicher Fehlzugriff beim Zwischenspeichern im Cache (Cache-Miss) wird durch den Ablauf einer Fehlerbehandlungsroutine in der Cache-Miss-Reparatur Logik behoben.

Abschließend wird das gültige VLIW am Ausgang des Decoders bereitgestellt.

Die Erfindung soll anhand eines Ausführungsbeispiels näher erläutert werden.

Die Fig. 1 zeigt eine Blockübersicht in dem die Kompressionsschritte dargestellt werden, welche in der Konfigurationsphase zur erfindungsgemäßen Wandlung des TVLIW 1 in das HVLIW 10 in der Betriebsart "Command Code" auszuführen sind.

In Fig. 2 ist eine Blockübersicht des erfindungsgemäßen Decoder 23 dargestellt, der in der Ausführungsphase das komprimierte HVLIW 10 in der Betriebsart "Command Code" in das VLIW 22 dekomprimiert und dekodiert, um damit den Prozessor 21 anzusteuern.

Wie in Figur 1 ersichtlich, ist in der Konfigurationsphase der Ausgangspunkt für die erfindungsgemäße Komprimierung das Vorliegen des TVLIW 1. Dieses besteht im Ausführungsbeispiel aus ersterem und zweitem TVLIW Container 11; 12.

Der jeweilige erste bzw. zweite TVLIW Container 11; 12 liegt mit seinen Bestandteilen, dem ersten bzw. zweiten Operationscode 2; 5, der ersten bzw. zweiten Kennung 3; 6 und dem ersten bzw. zweiten FIW 4; 7, vor.

In der auftretenden Reihenfolge wird jeweils ein TVLIW Container dem Code-Umsetzer 18 zugeführt und gleichzeitig wird im Code-Analyzer 8 die Kombination der drei Bestandteile eines TVLIW Containers auf ihre Häufigkeit ihres Auftretens in Bezug auf die weiteren TVLIW Container des jeweiligen TVLIW 1 durch Vergleich mit den Angaben des Wörterbuches 9 ermittelt.

Diese Angaben werden dem Code-Umsetzer 18 zur Verfügung gestellt. Dieser codiert daraus je nach der vorgesehenen Be-

triebsart den General Header 13 und verknüpft ihn mit den jeweiligen ersten bzw. zweiten FIW 4; 7, welche von den hintereinander bereitgestellten ersten bzw. zweiten TVLIW Containern 11; 12 entnommen werden.

5

Wenn alle TVLIW Container des TVLIW 1 abgearbeitet sind, wird der strukturierte General Header 13 bereitgestellt und liegt in den Teilen Header Mode 14, FU-C-Information 15, CE-Information 16 und Zusatzinformation 17 vor. Der General Header 13 wird der Folge von ersten bzw. zweiten FIW 4; 7 vorangestellt. Damit wird ein nun vollständiges HVLIW 10 im Speicher abgelegt.

10

Nachfolgend kann ein weiteres TVLIW 1 komprimiert werden.

15

Das Ende der erfindungsgemäßen Kompression ist erreicht, wenn alle TVLIW 1 in ein jeweiliges HVLIW 10 gewandelt sind.

20

Wie in Figur 2 ersichtlich, wird in der Ausführungsphase nach dem Zwischenspeichern (fetch) der Instruktionen und dadurch mit der Bereitstellung des HVLIW 10 und Decodieren dessen Header Mode 14 mit der vorliegenden Betriebsart "Command Code", der Einsatz des erfindungsgemäßen Decoder1 (23) zum Dekomprimieren/Dekodierens des HVLIW 10 ausgelöst.

25

Nachfolgend wird der General Header 13 als Bestandteil des HVLIW 10 in seinen Bestandteilen im Cache 26 zwischengespeichert und mittels dem Header Decoder 24 decodiert.

30

Zuerst wird mittels dem ersten Teil des General Header 13, dem Header Mode 14, seine Betriebsart erkannt und der Decoder1 23 entsprechend eingestellt.

35

Aus dem zweiten Teil des General Header 13, der FU-C-Information 15, wird die Informationen für die erste bzw. zweite FU 19; 20 entnommen, welche der ersten bzw. zweiten FIW 4; 7 von der CMDT 25 berücksichtigt werden müssen.

5 Aus dem dritten Teil des General Header 13, der CE-Information 16 wird der zu berücksichtigende Bereich der CMDT 25 verarbeitet. Aus dem letzten Teil des General Header 13 wird die Zusatzinformationen 17 entnommen.

Ein evtl. erkannter Fehlzugriff des Cache 26 wird von der Cache-Miss-Repair-Logik 27 behoben.

10 Mit diesen Informationen wird das VLIW 22 aufgebaut, indem die jeweiligen ersten und /oder zweiten FIW 4;7 in der dekodierten Reihenfolge und Position, in der die erste bzw. zweite FU 19; 20 entsprechend am Prozessor 21 nachfolgend angesteuert wird, im VLIW-22 angeordnet werden.

15

LIPPERT, STACHOW, SCHMIDT & PARTNER

Patentanwälte · European Patent Attorneys · European Trademark Attorneys

Krenkelstraße 3 · D-01309 Dresden

Telefon +49 (0) 3 51 3 18 18-0

Telefax +49 (0) 3 51 3 18 18 33

Ad-wb/wb

5. Juni 2002

5 **Systemonic AG****01099 Dresden**10 **Verfahren zum codieren/decodieren von VLIW cached Befehlen****Bezugszeichenliste**

-
- | | | |
|----|----|---|
| | 1 | TVLIW (Tagged Very Long Instruction Word) |
| 15 | 2 | erster Operationscode |
| | 3 | erste Kennung |
| | 4 | erstes FIW (Funktions-Instruktions-Wortteil) |
| | 5 | zweiter Operationscode |
| | 6 | zweite Kennung |
| 20 | 7 | zweites FIW |
| | 8 | Code-Analyzer |
| | 9 | Wörterbuch |
| | 10 | HVLIW (Headed Very Long Instruction Word) |
| | 11 | erster TVLIW Container |
| 25 | 12 | zweiter TVLIW Container |
| | 13 | General Header |
| | 14 | Header Mode |
| | 15 | FU-C-Information (FunctionUnit-Combination-Information) |
| | 16 | CE-Information (Cache Extra-Information) |
| 30 | 17 | Zusatzinformation |
| | 18 | Code-Umsetzer |
| | 19 | erste FU (Funktionseinheit) |
| | 20 | zweite FU (Funktionseinheit) |
| | 21 | Prozessor |
| 35 | 22 | VLIW (Very Long Instruction Word) |

- 23 Decoder1
 - 24 Header-Decoder
 - 25 CMT (Command-Code-Decompression-Tabelle)
 - 26 Cache
 - 5 27 Cache-Miss-Repair-Logik
-

LIPPERT, STACHOW, SCHMIDT & PARTNER

Patentanwälte · European Patent Attorneys · European Trademark Attorneys

Krenkelstraße 3 · D-01309 Dresden

Telefon +49 (0) 3 51 3 18 18-0

Telefax +49 (0) 3 51 3 18 18 33

Ad-wb/wb

5. Juni 2002

5 **Systemonic AG**01099 Dresden10. **Verfahren zum codieren/decodieren von VLIW cached Befehlen****Patentansprüche**

1. Verfahren zur Ansteuerung von Funktionseinheiten in einem
15 Prozessor, wobei in einer Konfigurationsphase eine aus einer Übersetzung eines Programmcodes stammende Folge von Primärinstruktionsworten eine Fraktionierung in eine Folge von Instruktionswortteile erfährt, wobei in einem Programmablauf den Prozessor ansteuernde Instruktionsworte in
20 der vollen Instruktionswortbreite zu einem VLIW aufgebaut und in einem Instruktionswortspeicher (Cache) zwischengespeichert werden, dadurch gekennzeichnet, dass die Fraktionierung eines Primärinstruktionswortes in der Konfigurationsphase, in einem ersten Schritt in die Folge einer bestimmten Anzahl von Instruktionswortteilen, die in
25 der Ausführungsphase für den Aufbau eines jeweiligen VLIW verwendet werden, erfolgt, wobei einem jeweiligen ersten und zweiten FIW (Funktions-Instruktions-Wortteil) (4), (7) ein ihm zugehöriger erster oder zweiter Operationscode (2), (5) vorangestellt wird, der damit bestimmt, wie in der Ausführungsphase mit dem durch das jeweilige FIW belegten Speicherplatz des Cache umgegangen wird,
30 dass sich dem jeweiligen ersten oder zweiten Operationscode (2), (5) eine zugehörige erste oder zweite Kennung (3), (6) anschließt, die die Information darstellt, welche erste oder zweite FU (19), (20) das jeweilige FIW ansteuert,
- 35

dass die ersten oder zweiten Operationscode (2), (5) und ihre zugehörigen ersten oder zweiten Kennung (3), (6) mit den jeweiligen ersten und zweiten FIW (4), (7) zu dem ersten bzw. zweiten TVLIW Container (11), (12) zusammengefasst werden, wobei diese in ihrer Gesamtheit das TVLIW (1) repräsentieren,

dass in einem zweiten Schritt in der Konfigurationsphase das jeweils vorliegenden TVLIW (1) in ein HVLIIW (10) gewandelt wird, wobei in dem HVLIIW (10) ein einvorangestellter General Header (13) enthalten ist, dass das HVLIIW (10) mit seiner codeverdichteten Struktur alle Funktionen des TVLIW (1) ersetzt.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass

eine Betriebsart "Command Code" des HVLIIW (10) und seines zugehörigen General Headers (13) realisiert wird,

dass in unmittelbarer Reihenfolge an den General Header (13) sich die für den Aufbau des VLIW (22) notwendigen ersten und zweiten FIW (4), (7) anschließen,

dass in diesem General Header (13) die Information codiert abgelegt ist, die alle Kombinationen angibt, welches erste und zweite FIW (Instruktionswortteil) (4), (7), nach Decodierung, in der Ausführungsphase, zur Ansteuerung einer jeweiligen ersten und/oder zweiten FU (Funktionseinheit) (19), (20) des Prozessors (21) vorgesehen ist,

dass im General Header (13) abgelegt ist, welches erste und/oder zweite FIW (4), (7) Speicherplätze des Cache (26) belegt und ob bzw. welche Operation mit dem jeweiligen Speicherinhalt bei Aufbau des VLIW (22) in der Ausführungsphase im Cache (26) ausgeführt werden sollen.

3. Verfahren nach Anspruch 1 und 2, dadurch gekennzeichnet, dass

in dem ersten Teil des General Headers (13) ein Header Mode (14), welcher eine Information über die Betriebsart "Command Code" des HVLIIW (10) und des General Headers (13) enthält, bereitgestellt wird,

dass diesem sich ein zweiter Teil anschließt, in dem als

Tabellenwerte codiert, die jeweilige meist gebrauchten Kombination abgelegt ist, welche der jeweiligen FU durch welches erstes oder zweites FIW (3), (7) angesteuert wird, dass ein dritter Teil als CE-Information (16), angeschlossen wird, in dem ein Zeiger, der auf eine vorgesehene Stelle des Wörterbuches (9) verweist, enthalten ist, dass als letzter Teil des General Headers (13) die Zusatzinformation (17) vorgesehen ist..

- 5 4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass eine Betriebsart "Referenz-Instruktion" des HVLIW (10) und des enthaltenen General Headers (13) realisiert wird, bei der die zum Aufbau des VLIW (22) in der Ausführungsphase vorgesehenen FIW im Cache (26) zwischengespeichert werden, wobei der zugehörige Header-Mode (14) eine entsprechend dekodierbare Kennung dieser Betriebsart "Referenz-Instruktion" trägt, dass die Betriebsart "Referenz-Instruktion" durch ein spezielles HVLIW (10) eingeleitet wird, in dem eine Adressangabe enthalten ist, mit der auf eine Referenz-Instruktion verwiesen wird, dass im nachfolgenden HVLIW (10), welches ebenfalls die Kennung der Betriebsart "Referenz-Instruktion" trägt, eine relative Adresse bezüglich der durch den Verweis vorliegenden Adressangabe enthalten ist, dass daran eine Maske der von der Ansteuerung auszuschließenden FU angefügt wird.

5. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass das spezielle HVLIW (10), welches die Betriebsart "Referenz-Instruktion" einleitet, mit seiner Adressangabe auf das vorhergehende HVLIW (10) verweist.

6. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass das spezielle HVLIW (10), welches die Betriebsart "Referenz-Instruktion" einleitet, mit seiner Adressangabe auf eine allgemeine Adresse verweist.

7. Verfahren nach Ansprüchen 1 bis 3, dadurch gekennzeichnet,
dass in der Ausführungsphase das HVLIIW (10) in einem Decoder1 (23) decodiert wird, welcher mit einem Header-Decoder (24), ein CMDT (25), einem Cache (26) und eine Cache-Miss-Reparatur Logik (27) ausgestattet ist, wobei das HVLIIW (10) im Cache (26) zwischengespeichert wird und
der Header-Decoder (24) die Betriebsart des General-Headers (13) aus dem dort gespeicherten Header Mode (14) erkennt,
dass entsprechend des erkannten Header Mode (14) die im General Header (13) vorliegenden Werte der FU-C-Information (14) durch Vergleich mit der CMDT (25) und in Verbindung mit der ebenfalls aus dem General Header (13) entnommenen CE-Information (16) dekomprimiert werden,
~~dass entsprechend des erkannten Header Modes (14) die Zusatzinformation (17) des General Headers (13) verarbeitet werden,~~
dass ein möglicher Fehlzugriff beim Zwischenspeichern im Cache (26) (Cache-Miss) durch den Ablauf einer Fehlerbehandlungsroutine in der Cache-Miss-Reparatur-Logik (28) behoben wird,
dass das gültige VLIW (22) am Ausgang des Decoder1 (23) bereitgestellt wird.

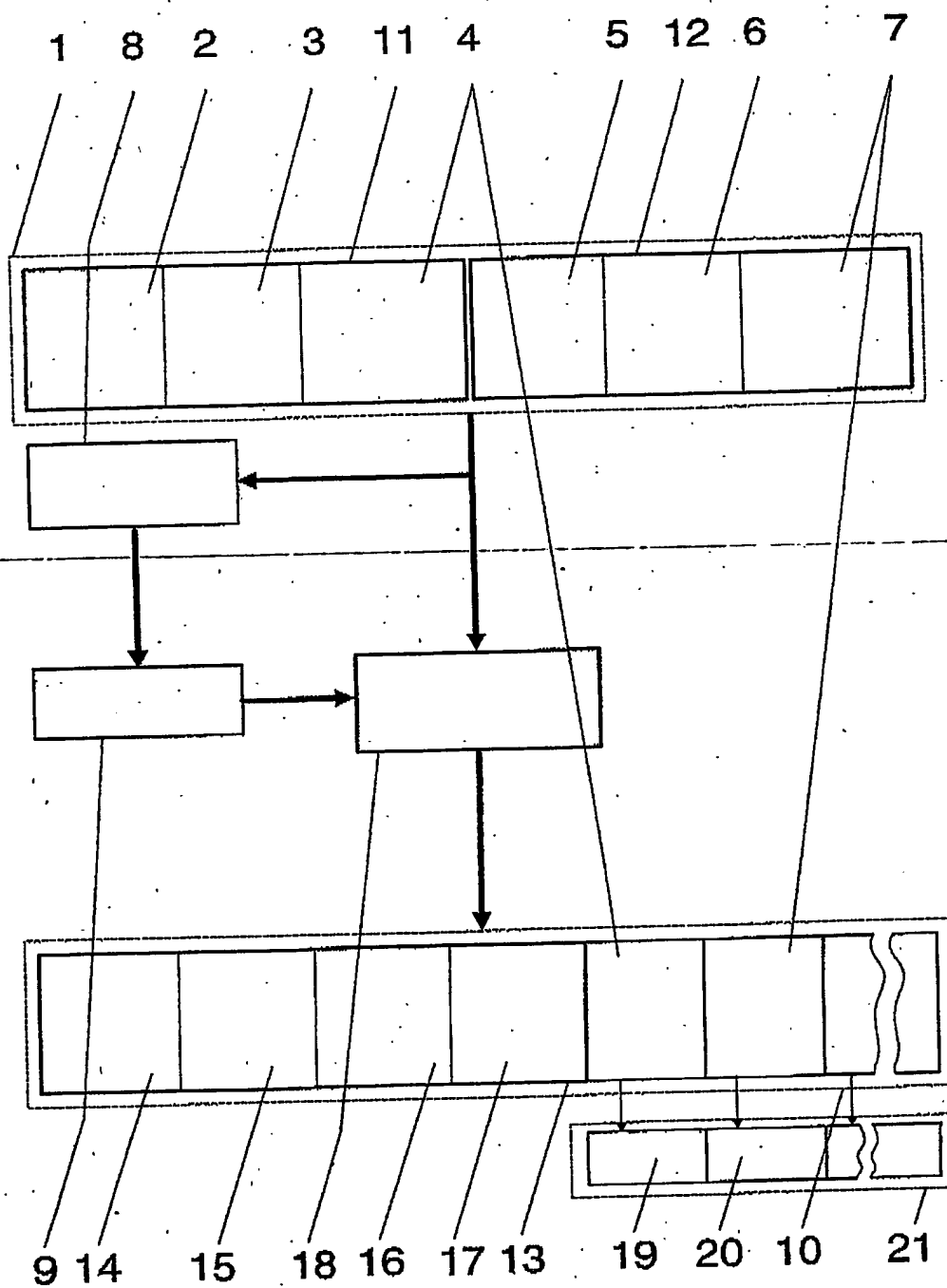


Fig. 1

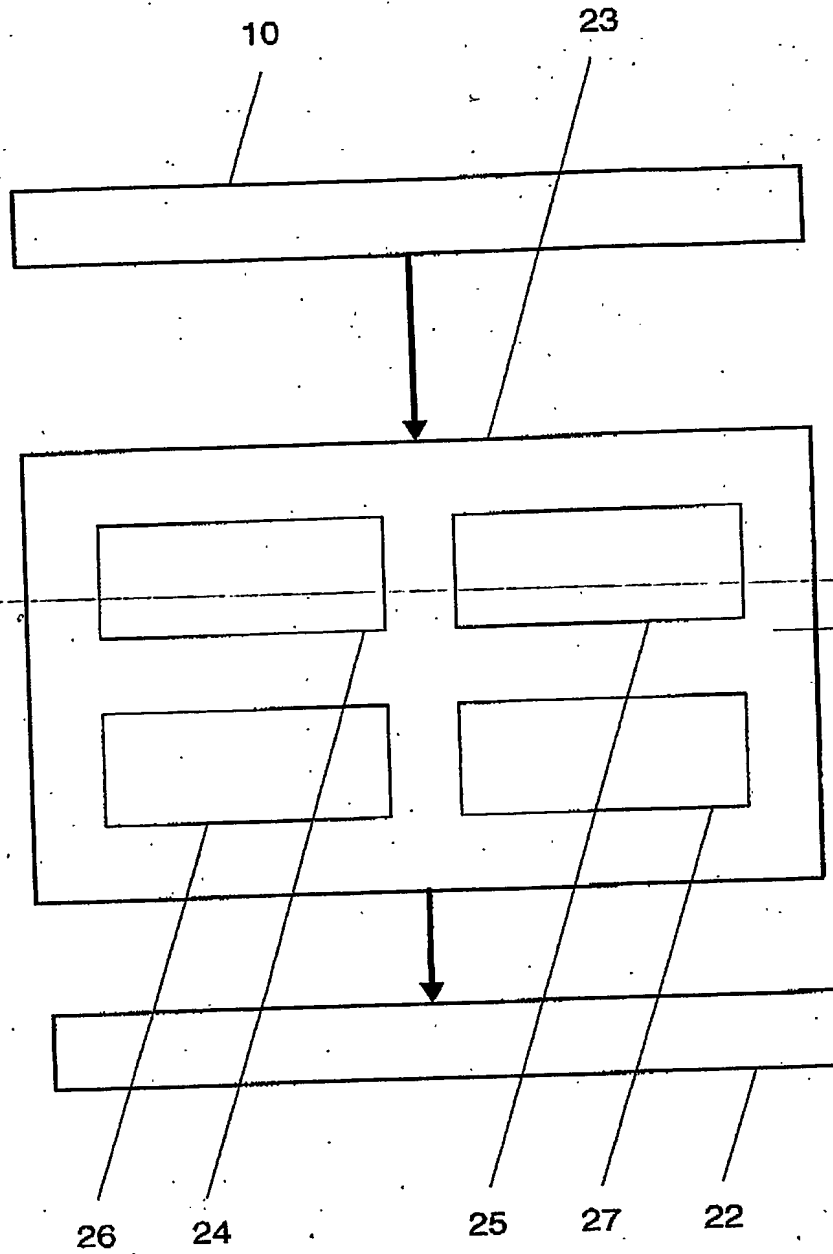


Fig. 2

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.